Computers & Fluids 108 (2015) 107–115

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Analysis and application of high order implicit Runge–Kutta schemes to collocated finite volume discretization of the incompressible Navier–Stokes equations

V. Kazemi-Kamyab *, A.H. van Zuijlen, H. Bijl

Aerodynamics Section, Faculty of Aerospace Engineering, Delft University of Technology, P.O. Box 5058, 2600 GB Delft, The Netherlands

ARTICLE INFO

Article history:
Received 22 July 2013
Received in revised form 31 October 2014
Accepted 20 November 2014
Available online 26 November 2014

Keywords:
Incompressible Navier–Stokes
High order time integration
Implicit Runge–Kutta
Collocated finite volume
Rhie–Chow interpolation

ABSTRACT

The application of a family of high order implicit Runge–Kutta time integration schemes, namely the explicit first-stage singly diagonally implicit Runge–Kutta schemes (ESDIRK), to cell-centered collocated finite volume discretization of the unsteady incompressible Navier–Stokes is considered. Although achieving computational efficiency relative to commonly used second order implicit schemes has been the motivating factor, this study focuses on temporal order analysis of the high order schemes on the collocated grid. In particular, a face velocity interpolation procedure (Rhie–Chow) which preserves the temporal design order of the ESDIRK schemes is introduced. The details of an iterative pressure-based time advancing algorithm comprising the designed interpolation method are presented (iterated-PISO). The results from solving numerical examples demonstrate the temporal order preservation of the algorithm.

1. Introduction

The unsteady incompressible Navier–Stokes equations in primitive variables are given by:

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\nabla \cdot (\boldsymbol{u}\boldsymbol{u}) + \nu \nabla^2 \boldsymbol{u} - \nabla p = R(\boldsymbol{u}) - \nabla p = \mathcal{F}(\boldsymbol{u}, p, t), \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0.$$

where $\boldsymbol{u}$ is the velocity vector, p is the kinematic pressure and ν the kinematic viscosity. The stability and accuracy of a numerical approach to solve (1) in primitive variables are influenced by the method with which the velocity–pressure coupling is resolved, both in the solution algorithm and in the arrangement of the variables [1].

The cell-centered collocated grid finite volume method (with an appropriate form of the Rhie–Chow procedure [2] to damp spurious pressure field) currently remains a popular choice for spatial discretization of engineering applications. For time integration, typically a fully implicit method is used in order to circumvent time step restrictions due to probable stiffness in the problem. Stiffness in a system can for example arise due to the nature of the governing equations or due to the generated grid (such as clustering of nodes near an area of interest [3]). The commonly used time integration schemes are the second order Crank–Nicolson and second order backward difference (BDF2) schemes. The resultant fully discretized equations are then typically solved using a segregated solution algorithm (e.g. SIMPLE-like or PISO-like methods). However, performing unsteady computations can become computationally expensive even for engineering accuracy levels. As a potential solution for reducing the computational cost, high order implicit time integration schemes have been proposed in the literature to be employed in place of the commonly used second order schemes. For example, in [4] computational efficiency of a class of high order multi-stage time integration schemes, Rosenbrock–Wanner (ROW), relative to Crank–Nicolson is demonstrated for an incompressible flow test-case. An inf–sup stable (Q_2/P_1) finite element method is used for spatial discretization and the algebraic system of the fully discretized INS equations is solved directly (i.e. a segregated solution method is not used). Another example can be found in [5], for a compressible flow test-case, where the computational efficiency of high order explicit first-stage singly diagonally implicit Runge–Kutta (ESDIRK) schemes relative to BDF2 has been demonstrated.

In this paper we apply the high order ESDIRK schemes to the cell-centered collocated finite volume discretization of the INS equations, and use a (PISO-like) segregated solution algorithm to solve the resultant fully discretized equations. However, without considering special treatments the obtained solution is susceptible

* Corresponding author.
E-mail address: v.kazemikamyab@tudelft.nl (V. Kazemi-Kamyab).

http://dx.doi.org/10.1016/j.compfluid.2014.11.025
0045-7930/

to temporal order reduction (in the asymptotic range). Order reduction will result in the decrease or loss of the potential gain in computational efficiency from using the high order time integration schemes compared to the lower order schemes. Therefore, this study focuses on temporal order analysis of the high order ESDIRK schemes on the collocated grid. In particular, it presents a method to ensure the solution algorithm preserves the temporal design order of the applied high order time integration schemes. Here, this property is referred to as temporal order preservation of the solution algorithm.

It is well known that a standard collocated grid finite volume method has convergence problems and spurious oscillations in the pressure field can occur which deteriorate the solution [1,6]. The checker-board pressure field arises as result of centered discretization of the continuity equation and the pressure gradient in the momentum equation [1,6]. A widely used method in the literature and CFD packages which tackles this problem is the interpolation procedure proposed by Rhie and Chow [2]. The basic mathematical principle of this method is to suppress the spurious pressure modes by perturbing the continuity equation with a pressure regularization (dissipation) term [7]. In this approach discretized momentum equations are formulated for the convective (or face) velocities which appear in the discrete continuity equation [1]. The result of the interpolation technique is an expression for the face velocity which consists of the standard linear interpolation of the cell-centered velocities with the addition of a pressure smoothing term [8,6,9]. It can be shown, with certain assumptions, that the pressure smoothing term is a third order artificial dissipation term, which accordingly results in the introduction of a fourth order dissipation term in the continuity equation [6,9]. In addition, since the face velocity enters the convective flux of the momentum equation, the interpolation also introduces a fourth-order smoothing term into the momentum equation [9].

A defect of this interpolation procedure introduced for steady-state computations is that its extension to unsteady flow conditions is not obvious [9]. In computing the solution to the unsteady INS equations (in particular using a class of segregated solution algorithms such as SIMPLE-like and PISO-like approaches), care should be taken in selecting the interpolation method, otherwise it will result in a time-step dependent steady-state solution, temporal order reduction, and/or temporal inconsistency. A recent review of the literature on this topic is presented in [10]. To the authors' knowledge, currently a few time-consistent interpolation schemes are available in the literature for Backward Euler (BDF1); they are the interpolation schemes proposed by Lien and Leschziner [9] or similarly Yu et al. [11], Cubero and Fueyo [8], and Pascau [10]. It follows that in applying high order implicit time schemes, one of the three interpolation procedures should be adopted. Yet, additional considerations might be required in order to preserve temporal consistency and/or order of the higher order schemes. For example, for the commonly used second order schemes, these issues have been addressed and studied in [8,11,12]. In this paper, a face velocity interpolation procedure which preserves the temporal design order of the high order multi-stage ESDIRK schemes is introduced. In addition, the details of an iterative pressure-based time advancing algorithm comprising the designed interpolation method are discussed.

In what follows, first the semi-discrete form of (1) is obtained by discretizing the spatial operators. After a brief discussion on high order ESDIRK schemes, the details of a corresponding temporally order-preserving face-velocity interpolation method is presented. This is followed by discussing the solution algorithm. Next, numerical examples are used to investigate the temporal order preservation of the algorithm.

2. Spatial discretization and semi-discrete form

The method of lines [13] is used to discretize (1), with spatial operators being discretized first to obtain the semi-discrete form of the equations. The cell-centered finite volume approach is considered for spatial discretization with collocated arrangement of the primitive variables. The computational domain is subdivided into finite volumes (or cells) where each cell is bounded by arbitrary number of cell faces. The volume integral form of (1) is applied to each finite volume, and the Gauss divergence theorem is used to convert the volume integrals to surface integration over the closed boundary of the cell (see [12,14] for more details). The resultant discrete form of (1) is given by:

$$V_P \frac{d\boldsymbol{U}_P}{dt} = -\sum_f \phi_f \boldsymbol{U}_f + \nu \sum_f (\nabla \boldsymbol{U})_f \cdot \boldsymbol{n}_f S_f - \sum_f p_f \boldsymbol{n}_f S_f, \tag{2}$$

$$\sum_f \boldsymbol{U}_f \cdot \boldsymbol{n}_f S_f = 0. \tag{3}$$

where the subscripts P and f represent the cell-center and face-center values (see Fig. 1). V_P is the volume of the cell, and for a given cell face f, $\boldsymbol{n}_f$ is the face normal vector, and S_f is its surface area. The convective velocity through the face is denoted by $\boldsymbol{U}_f$, and $\phi_f = \boldsymbol{U}_f \cdot \boldsymbol{n}_f S_f$ is the mass flow rate or flux through the face.

After using appropriate schemes to discretize the spatial operators, the following semi-discretized form of (1) for each cell P is obtained (where (2) is divided by the cell volume):

$$\frac{d\boldsymbol{U}_P}{dt} + a_P \boldsymbol{U}_P - \sum_{N=nb(P)} a_N \boldsymbol{U}_N = -(\nabla p)_P + \boldsymbol{r}_P, \tag{4}$$

$$\sum_f^{N_f} \phi_f = 0 \tag{5}$$

where a_P and a_N are the diagonal and off-diagonal coefficients of the (spatial) discretization matrix. $\boldsymbol{r}_P$ includes parts of convection and diffusion terms treated explicitly, as well as boundary contributions. The term $(\nabla p)_P$ denotes the (average) pressure gradient at P, $(\nabla p)_P = \frac{1}{V_P}\sum_f^{N_f} p_f \boldsymbol{n}_f S_f$ (Green-Gauss gradient reconstruction). The pressure at the face is evaluated using linear interpolation of the adjacent cell-centered values. Using the following definition:

$$\boldsymbol{H}_P(\boldsymbol{U}) = \sum_{N=nb(P)} a_N \boldsymbol{U}_N + \boldsymbol{r}_P, \tag{6}$$

we can rewrite the semi-discretized form of the momentum equation in (4) as:

$$\frac{d\boldsymbol{U}_P}{dt} + a_P \boldsymbol{U}_P = \boldsymbol{H}_P - (\nabla p)_P. \tag{7}$$

By defining the residual vector $\boldsymbol{\mathcal{F}}_P$ as,

$$\boldsymbol{\mathcal{F}}_P = -a_P \boldsymbol{U}_P + \boldsymbol{H}_P - (\nabla p)_P, \tag{8}$$

Eq. (7) can be expressed by,

$$\frac{d\boldsymbol{U}_P}{dt} = \boldsymbol{\mathcal{F}}_P(\boldsymbol{U}, \boldsymbol{p}, t). \tag{9}$$

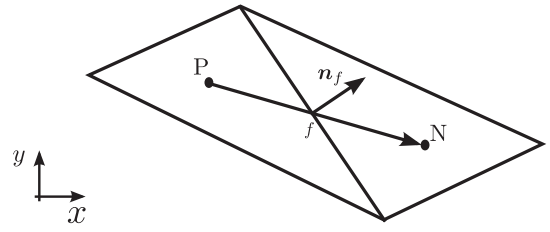

Fig. 1. Geometric sketch of a control volume and its neighboring cell.

3. Time integration

In this paper, the high order ESDIRK schemes are considered for time integration which can be made of arbitrary high order while retaining the L-stability property. While BDF1 and BDF2 are L-stable, the Crank–Nicolson scheme is only A-stable. It is possible to construct higher order BDF methods, but they are only $L(\alpha)$ stable and their stability region drops as the order of the scheme is increased (for further details see [15,16]). For an ODE system of the form $\frac{d\boldsymbol{U}_P}{dt} = \boldsymbol{\mathcal{F}}_P(\boldsymbol{U}, \boldsymbol{p}, t)$, the solution at each stage of the ESDIRK scheme can be written as:

$$\frac{\boldsymbol{U}_P^{(k)} - \boldsymbol{U}_P^n}{\Delta t} = \sum_{i=1}^{k} a_{ki} \boldsymbol{\mathcal{F}}_P(\boldsymbol{U}^{(i)}, \boldsymbol{p}^{(i)}, t^n + c_i \Delta t), \tag{10}$$

or,

$$\frac{\boldsymbol{U}_P^{(k)} - \boldsymbol{U}_P^n}{a_{kk}\Delta t} = \boldsymbol{\mathcal{F}}_P^{(k)} + \frac{1}{a_{kk}} \sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_P^{(i)}, \tag{11}$$

where a_{ki} are the stage weights, and $c_i = \sum_j a_{ij}$ are the quadrature nodes of the scheme. The solution at the next time level is obtained by,

$$\boldsymbol{U}_P^{n+1} = \boldsymbol{U}_P^n + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{\mathcal{F}}_P^{(i)}, \tag{12}$$

where b_i are the scheme's main weights with $\sum_i b_i = 1$, and s is the number of stages. In this paper, stiffly accurate ESDIRK schemes are considered ($a_{si} = b_i$ and $c_s = 1$) which forces the solution at the new time level to be computed implicitly, i.e. the solution of the last stage is equal to the solution of the new time-level, $\boldsymbol{U}_P^{n+1} = \boldsymbol{U}_P^{(s)}$. Therefore, computing (12) becomes unnecessary. The coefficients and weights are usually arranged in a Butcher tableau (see Fig. 2). For the ESDIRK schemes, as the name implies, the diagonal coefficients are equal ($a_{kk} = \gamma$). As Fig. 2 shows, the first stage of the ESDIRK scheme is explicit which allows the scheme to have a stage order of two as opposed to the stage order of one for the more traditional SDIRK schemes [17]. Since the ESDIRK schemes posses stages with lower orders of accuracy (at least a stage order of two) compared to their design order, they are susceptible to order-reduction in the presence of substantial stiffness (see [5,18,15]). In this paper, the high order ESDIRK schemes developed in Kennedy and Carpenter [17] are used for time integration. The schemes have been optimized in [17] to minimize both leading order error terms of the schemes and Butcher coefficient magnitudes as well as maximize conservation properties. It is also possible to optimize the Runge–Kutta schemes for low dissipation and dispersion. The interested reader is referred to [19,20] for more on this topic.

4. Face-velocity interpolation

In order to avoid pressure–velocity decoupling, the method proposed by Rhie and Chow [2] is followed here; rather than evaluating the face velocities $\boldsymbol{U}_f$ from a direct interpolation of the neighboring cell-center nodes, fully-discretized momentum equation for the face velocities are constructed. For unsteady flow computation with BDF1 for time integration, there are a few

c_1	0	0	0	0
c_2	a_{21}	a_{22}	0	0
c_3	a_{31}	a_{32}	a_{33}	0
c_4	a_{41}	a_{42}	a_{43}	a_{44}
	b_1	b_2	b_3	b_4

Fig. 2. A four stage ESDIRK scheme.

time-consistent interpolation schemes available in the literature as discussed in the introduction. Here, the approach of Lien and Leschziner [9] (or similarly Yu et al. [11]) is adopted to evaluate the appropriate terms in the discretized momentum equation formulated for the face-velocity.

As (11) shows, from an implementation view point, each stage of the ESDIRK scheme resembles the BDF1 scheme with a source term ($\frac{1}{a_{kk}} \sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_P^{(i)}$). We will proceed with deriving an equation for the face velocity with particular emphasis on the treatment of the source term (which arises from time discretization) such that the design order of the ESDIRK schemes is preserved.

Applying the ESDIRK schemes to the semi-discrete form of the momentum Eq. (7), we arrive at (with reference to (11)),

$$\tilde{a}_P \boldsymbol{U}_P^{(k)} = \frac{1}{\Delta t a_{kk}} \boldsymbol{U}_P^n - (\nabla p)_P^{(k)} + \boldsymbol{H}_P^{(k)} + \frac{1}{a_{kk}} \sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_P^{(i)}, \tag{13}$$

where

$$\tilde{a}_P = \frac{1}{\Delta t a_{kk}} + a_P. \tag{14}$$

Dividing (13) by $\tilde{a}_P$, yields the equation for the cell-center velocity:

$$\boldsymbol{U}_P^{(k)} = \frac{1}{\Delta t a_{kk}} \frac{\boldsymbol{U}_P^n}{\tilde{a}_P} + \frac{\boldsymbol{H}_P^{(k)}}{\tilde{a}_P} - \frac{(\nabla p)_P^{(k)}}{\tilde{a}_P} + \frac{1}{a_{kk}} \frac{\sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_P^{(i)}}{\tilde{a}_P}. \tag{15}$$

An equivalent equation as the one for the centered velocity (15) can be defined for the face (convective) velocity:

$$\boldsymbol{U}_f^{(k)} = \frac{1}{\Delta t a_{kk}} \frac{\boldsymbol{U}_f^n}{\tilde{a}_f} + \frac{\boldsymbol{H}_f^{(k)}}{\tilde{a}_f} - \frac{(\nabla p)_f^{(k)}}{\tilde{a}_f} + \frac{1}{a_{kk}} \frac{\sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_f^{(i)}}{\tilde{a}_f}, \tag{16}$$

where $(\nabla p)_f$ denotes the face pressure gradient normal to the surface f. On a cartesian grid, the gradient is computed using the central differencing scheme. Following Lien and Leschziner [9], $\boldsymbol{H}_f$ and $\tilde{a}_f$ are approximated by the following interpolations:

$$\boldsymbol{H}_f = [\boldsymbol{H}_P]_f, \tag{17}$$

$$\tilde{a}_f = [\tilde{a}_P]_f, \tag{18}$$

where the symbol $[\zeta]_f$ denotes linear interpolation of the cell-centered values encompassing the face f. With reference to Fig. 1,

$$[\zeta]_f = f_x \zeta_P + (1 - f_x)\zeta_N, \tag{19}$$

where the interpolation coefficient f_x is defined by the ratio $\frac{\overline{fN}}{\overline{PN}}$ [14].

To retain the temporal consistency of the method, $\boldsymbol{H}_P$ in (6) only contains contributions resulting from the discretization of the spatial operators, and does not include the previous time-step solution of the cell-centered velocity $\boldsymbol{U}_P^n$ [8–10].

The face residual vector $\boldsymbol{\mathcal{F}}_f$ is evaluated based on considering the fully-discretized form of $\frac{d\boldsymbol{U}_f}{dt} = \boldsymbol{\mathcal{F}}_f$:

$$\frac{\boldsymbol{U}_f^{(k)} - \boldsymbol{U}_f^n}{\Delta t a_{kk}} = \boldsymbol{\mathcal{F}}_f^{(k)} + \frac{1}{a_{kk}} \sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_f^{(i)}. \tag{20}$$

Noting that at the end of a stage $\boldsymbol{U}_f^{(k)}$ is known, it is possible to evaluate $\boldsymbol{\mathcal{F}}_f^{(k)}$ using:

$$\boldsymbol{\mathcal{F}}_f^{(k)} = \frac{\boldsymbol{U}_f^{(k)} - \boldsymbol{U}_f^n}{\Delta t a_{kk}} - \frac{1}{a_{kk}} \sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_f^{(i)}. \tag{21}$$

The (cell-center) pressure is evaluated by transforming the discrete continuity Eq. (3) into an equation for pressure. Substituting (16) into (3) (assuming that $\boldsymbol{U}_f^{(k)}$ is divergence free), yields:

$$\sum_f \left(\frac{1}{\tilde{a}_f} (\nabla p)_f^{(k)} \right) \cdot \boldsymbol{n}_f S_f = \sum_f \left(\frac{1}{\Delta t a_{kk}} \frac{\boldsymbol{U}_f^n}{\tilde{a}_f} + \frac{\boldsymbol{H}_f^{(k)}}{\tilde{a}_f} + \frac{1}{a_{kk}} \frac{\sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_f^{(i)}}{\tilde{a}_f} \right) \cdot \boldsymbol{n}_f S_f. \tag{22}$$

By rewriting (13) as an equation for $\boldsymbol{H}_P$, and substituting it into (16), and by noting (17), an equivalent equation for the face velocity is obtained:

$$\boldsymbol{U}_f^{(k)} = \frac{\left[\tilde{a}_P \boldsymbol{U}_P^{(k)}\right]_f}{\tilde{a}_f} - \frac{(\nabla p)_f^{(k)} - \left[(\nabla p)_P^{(k)}\right]_f}{\tilde{a}_f} + \frac{1}{\Delta t a_{kk}} \frac{\boldsymbol{U}_f^n - [\boldsymbol{U}_P^n]_f}{\tilde{a}_f} + \frac{1}{a_{kk}} \frac{\sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_f^{(i)} - \left[\sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_P^{(i)}\right]_f}{\tilde{a}_f}, \tag{23}$$

where $\tilde{a}_f$ is evaluated by (18). By noting that at steady-state, $\boldsymbol{U}_m^{n+1} = \boldsymbol{U}_m^{(k)} = \boldsymbol{U}_m^n$, and $\boldsymbol{\mathcal{F}}_m = 0$ (with $m = P$ or f), (23) reduces to the following form:

$$\boldsymbol{U}_f = \frac{[a_P \boldsymbol{U}_P]_f}{a_f} - \frac{(\nabla p)_f - [(\nabla p)_P]_f}{a_f} \tag{24}$$

Therefore, the steady state solution is independent of Δt and the residual vectors $\boldsymbol{\mathcal{F}}_P$ and $\boldsymbol{\mathcal{F}}_f$.

5. Solution algorithm

There are several routes available for advancing the solution to the unsteady INS equations in time, of which the pressure or pressure correction methods (also including the non-iterative fractional step or projection methods) are commonly used approaches. These methods allow the system to be solved as a series of individual uncoupled advection–diffusion equations for each of the velocity components and as an equation for the pressure [21]. As a result of the segregated treatment of the equations however, the solution contains an additional source of temporal error, denoted as the splitting error [21,22]. In addition, by treating the non-linear convection term implicity (a fully implicit discretization), an iterative method is required to resolve the nonlinearities. Therefore, the solution contains three temporal error components: time integration error (discretization of the time derivative), splitting error, and non-linear iterative error [18]. Here, the solution field at each (implicit) stage of the ESDIRK schemes is computed using an iterated PISO algorithm [7,12] (no under-relaxation is considered in this paper). The PISO method is used to increase the rate of convergence of the iterations (and accordingly the rate at which the splitting error and the non-linear iterative error reduce) [7].

Within an implicit stage k of the ESDIRK schemes, and at each iteration j of the iterated PISO algorithm, the following steps are performed:

1. Solve the discretized momentum equation (predictor step). In order to proceed with the computations at each iteration j, previous iteration values are used for the variables which need to be guessed (i.e. pressure, mass-flux fields, etc.).

$$\tilde{a}_P \boldsymbol{U}_P^* - \sum_{N=nb(P)} a_N \boldsymbol{U}_N^* = \frac{1}{\Delta t a_{kk}} \boldsymbol{U}_P^n - (\nabla p^{j-1})_P + \boldsymbol{r}_P + \frac{1}{a_{kk}} \sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_P^{(i)}, \tag{25}$$

2. Calculate mass-fluxes (right hand side of the pressure Eq. (22)):

$$\phi_f^* = \left(\frac{1}{\Delta t a_{kk}} \frac{\boldsymbol{U}_f^n}{\tilde{a}_f} + \frac{\boldsymbol{H}_f}{\tilde{a}_f} + \frac{1}{a_{kk}} \frac{\sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_f^{(i)}}{\tilde{a}_f} \right) \cdot \boldsymbol{n}_f S_f, \tag{26}$$

where $\boldsymbol{H}_f$ and $\tilde{a}_f$ are computed according to (17) and (18) respectively.

3. Solve the equation for pressure.

$$\sum_f \left(\frac{1}{\tilde{a}_f} (\nabla p^j)_f \right) \cdot \boldsymbol{n}_f S_f = \sum_f \phi_f^*. \tag{27}$$

4. Correct the mass-flux, the convective and cell-center velocity fields.

$$\boldsymbol{U}_f^j = \frac{1}{\Delta t a_{kk}} \frac{\boldsymbol{U}_f^n}{\tilde{a}_f} + \frac{\boldsymbol{H}_f}{\tilde{a}_f} - \frac{(\nabla p^j)_f}{\tilde{a}_f} + \frac{1}{a_{kk}} \frac{\sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_f^{(i)}}{\tilde{a}_f}, \tag{28}$$

$$\phi_f^j = \boldsymbol{U}_f^j \cdot \boldsymbol{n}_f S_f, \tag{29}$$

$$\boldsymbol{U}_P^j = \frac{1}{\Delta t a_{kk}} \frac{\boldsymbol{U}_P^n}{\tilde{a}_P} + \frac{\boldsymbol{H}_P}{\tilde{a}_P} - \frac{(\nabla p^j)_P}{\tilde{a}_P} + \frac{1}{a_{kk}} \frac{\sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_P^{(i)}}{\tilde{a}_P}. \tag{30}$$

The PISO loop consists of performing steps 2–4 (at least) twice. At the beginning of each PISO loop, $\boldsymbol{H}_P(\boldsymbol{U})$ (which is used in the evaluation of $\boldsymbol{H}_f(\boldsymbol{U})$, (17)) is computed based on the current value of $\boldsymbol{U}_P$. Furthermore, within each inner (PISO) iteration, the coefficients of the discretization matrices ($\tilde{a}_P$, a_N, and accordingly $\tilde{a}_f$) remain constant, but change from one outer iteration to another.

As pointed out in Issa [23], each additional PISO-iteration increases the order of accuracy of the splitting error by one. However, noting that in each outer iteration j, solving the elliptic pressure equation is typically the most time consuming part, performing more PISO-loops to lower the splitting error does not seem to offer significant advantage over performing a complete outer iteration.

After performing sufficient iterations to resolve the nonlinearities in the momentum equation and the coupling between the equations to a specified tolerance, the cell-center and face residual vectors are computed using (8) and (21) respectively.

6. Numerical examples and discussion

Four test cases are considered in this section in order to assess the temporal order preservation of the presented algorithm.

6.1. Lid driven cavity flow

The first test case is a two-dimensional lid driven cavity flow with $Re = \frac{LU_{lid}}{\nu} = 10$ (U_{lid}: lid speed, L: height of the cavity, ν: fluid kinematic viscosity). The domain is a unit square surrounded by solid walls. The top wall is sliding to the right with a velocity of magnitude $U_{lid} = 1$. A uniform grid of 50×50 has been used. The no slip boundary condition is imposed for the velocity at the walls, and zero gradient for the pressure. The convective and diffusive terms are discretized using second order centered schemes. At each stage the linear system for each flow variable is solved to a tolerance of $\epsilon_{solver} < 10^{-15}$. In addition, the solution algorithm is iterated to a strict tolerance of $\epsilon_{iter} < 10^{-11}$ to eliminate *iteration* error (splitting error and non-linear iterative error) as a contaminating variable in the study. ϵ_{iter} is the L_2 norm of the magnitude of the residual vector $\boldsymbol{\mathcal{R}}$ of the fully-discretized momentum equation. The residual vector $\boldsymbol{\mathcal{R}}^j$ at each iteration has been computed using,

$$\boldsymbol{\mathcal{R}}^j = -\frac{\boldsymbol{U}_P^j - \boldsymbol{U}_P^n}{\Delta t a_{kk}} + \boldsymbol{\mathcal{F}}_P^j + \frac{1}{a_{kk}} \sum_{i=1}^{k-1} a_{ki} \boldsymbol{\mathcal{F}}_P^{(i)}. \tag{31}$$

Furthermore, at each iteration j, two PISO iterations are used.

A few definitions which will be used throughout this section are given here. The temporally exact solution $\boldsymbol{U}_{exact}$ is defined as one acquired by solving the problem using a fine time-step size. The time integration error $\boldsymbol{\epsilon}_t$ is defined as the difference between the temporally exact solution and the solution obtained using a coarse time-step size relative to the temporally exact solution $\boldsymbol{\epsilon}_t = \boldsymbol{U}_{exact} - \boldsymbol{U}$ (assuming that the iteration error is negligible).

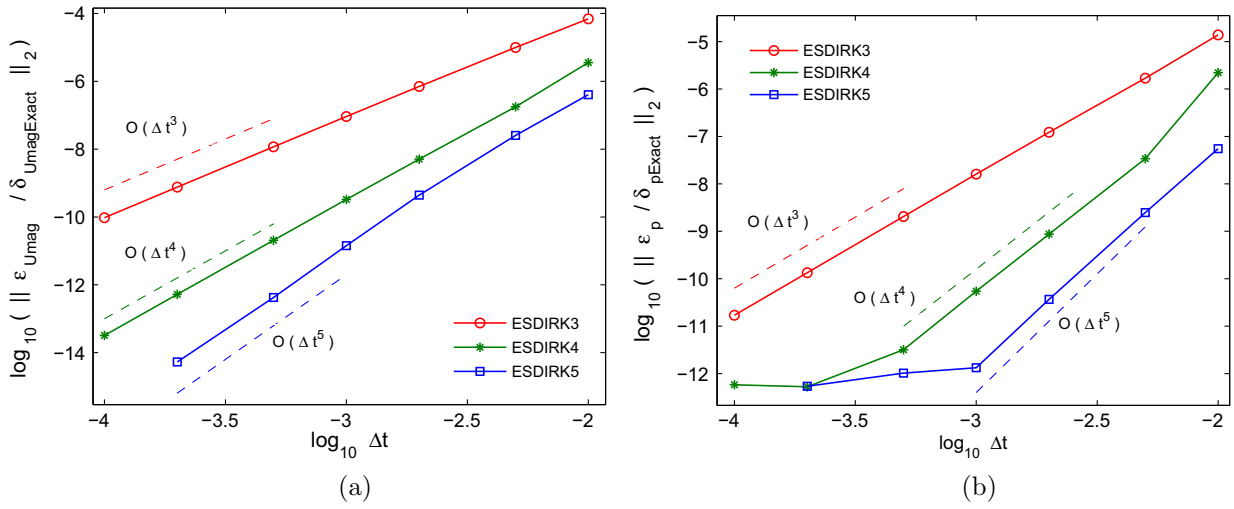

Fig. 3. Error ε as a function of Δt for (a) cell-center velocity field magnitude U_{mag} and (b) pressure field p. $\delta_\ell = |\max(\ell) - \min(\ell)|$.

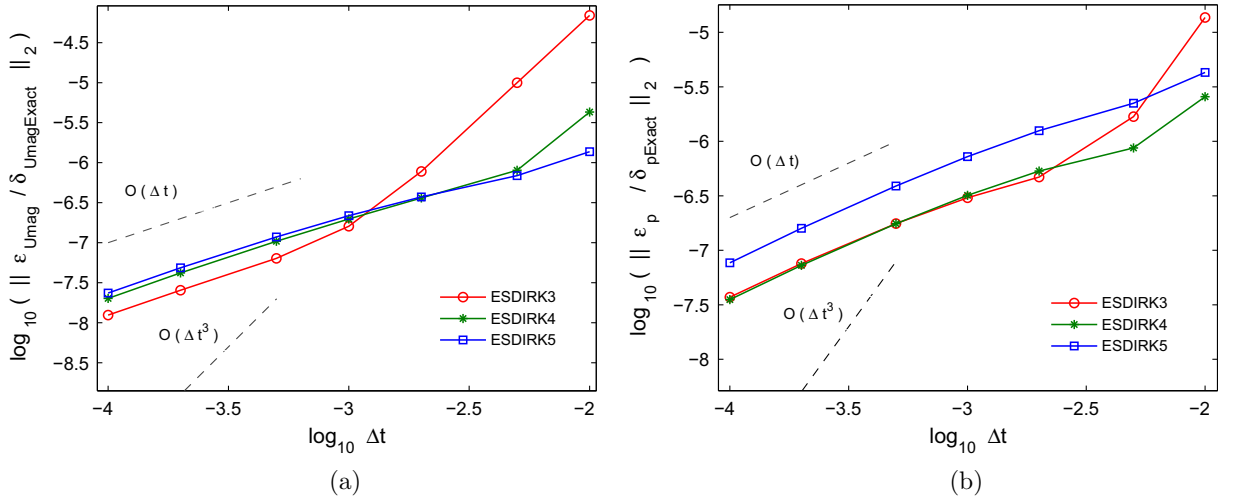

Fig. 4. Error as a function of Δt for (a) cell-center velocity field magnitude and (b) pressure field, for the algorithm where sum of the previously evaluated cell center residual vectors $\mathcal{F}_p$ is included in $\boldsymbol{H}_p$.

The temporal accuracy of the algorithm was investigated by considering the cavity flow start-up process. The fluid is initially at rest and computations are carried out to $t_{final} = 0.05$. The temporally exact solution is obtained using the fifth order ESDIRK (ESDIRK5) with time step-size of $\Delta t_{fine} = 1 \times 10^{-4}$. The (time integration) error is evaluated by comparing the results at t_{final}. The error in the cell-center velocity and pressure are shown as a function of time step size, in respectively Fig. 3(a) and (b). The maximum Courant number at t_{final} using ESDIRK-5 and $\Delta t = 0.01$ was $Co = 0.45$.

As the results demonstrate, the design orders of the 3rd–5th order ESDIRK schemes (denoted in the figures by ESDIRK3-ESDIRK5) are clearly observable in both the velocity and pressure fields. Furthermore, the pressure appears to be more sensitive (in comparison to the velocity field) to the imposed iterative tolerance ($\epsilon_{iter} < 10^{-11}$) and the accuracy of the solution does not increase by reducing the time-step size once the error in the solution reaches the tolerance.

Since $\frac{1}{a_{kk}} \sum_{i=1}^{k-1} a_{ki} \mathcal{F}_P^{(i)}$ in (15) does not explicitly depend on Δt, it might appear that it is possible to include it in $\boldsymbol{H}_p$ (i.e. $\mathcal{F}_f = [\mathcal{F}_P]_f$); hence in evaluating $\boldsymbol{U}_f$, there is no need for defining an additional variable $\mathcal{F}_f$. Temporal order analysis was also carried out for this solution algorithm. For each ESDIRK scheme, its solution at t_{final} obtained with a time step-size of $\Delta t_{fine} = 1 \times 10^{-5}$ was used as the temporally exact solution. As Fig. 4 demonstrates, by using this algorithm the solution accuracy reduces to first order in the asymptotic range.

6.2. Lid driven cavity flow-unstructured grid

For many geometrically complex engineering applications, the unstructured grid is preferred due to its flexibility. To demonstrate the order preservation of the algorithm on such grids, this time the cavity is meshed with an unstructured triangular mesh with 3448 vertices and 6678 cells as shown in Fig. 5. The Reynolds number is increased to $Re = \frac{LU_{lid}}{\nu} = 100$. Linear interpolation is used for discretizing the convection terms. In the momentum and pressure equations, the discretization of the face-normal gradient of respectively the velocity components and the pressure includes two terms (see [14,12] for more details): an orthogonal contribution which is treated implicitly, and a non-orthogonal correction which is treated explicit (lags behind by one iteration). In correcting the face velocity vector $\boldsymbol{U}_f^j$, (28), the face normal pressure gradient is evaluated by computing the orthogonal term using the updated

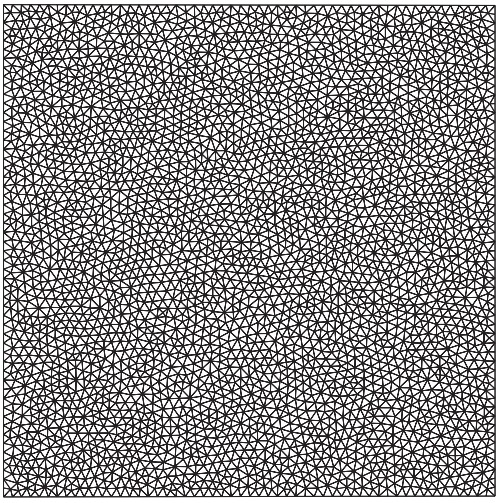

Fig. 5. Unstructured mesh for flow in cavity.

pressure field (p^j) and the non-orthogonal term using the pressure field at the previous iteration (p^{j-1}); this is necessary to ensure mass conservation, but may adversely affect temporal order in the asymptotic range. The temporal order analysis also examines this issue.

At each stage the following algorithm tolerances are imposed: $\epsilon_{solver} < 10^{-15}$, $\epsilon_{iter} < 10^{-12}$. Again 2-PISO iterations are performed in each outer iteration (j). Similar to the previous test case, the temporal accuracy of the algorithm is investigated by considering the cavity flow start-up process. The fluid is initially at rest and computations are carried out to $t_{final} = 0.5$. The temporally exact solution was obtained using the fifth order ESDIRK (ESDIRK5) with a time step-size of $\Delta t_{fine} = 10^{-4}$. The (time integration) error in the cell-center velocity and pressure are shown as a function of time step size, in respectively Fig. 6(a) and (b). The maximum Courant number at t_{final} using ESDIRK-5 with $\Delta t = 0.05$ is $Co = 5$.

As the results demonstrate, the design orders of the 3rd–5th order ESDIRK schemes are preserved on the unstructured mesh in both the velocity and pressure fields.

6.3. Oscillatory lid driven cavity flow

The advantage of using implicit time integration schemes with iterated solution algorithms lies in the ability to consider large Courant numbers. This is demonstrated here for the presented solution algorithm in which the high order ESDIRK schemes are used for time integration. For this analysis, an oscillatory lid driven cavity is considered, where a sinusoidal forcing function is imposed on the top wall,

$$U_{lid} = sin(\varpi t). \tag{32}$$

ϖ is the frequency of the oscillations, and is related to the period of the oscillation ($\mathcal{T}$) by $\mathcal{T} = \frac{2\pi}{\varpi}$. The periodicity of the flow provides a good perception of whether a certain time-step is coarse in resolving the periodic solution. This test case has been studied for example in [24–26]. Here, simulations are performed for $Re = 400$ and frequency of $\varpi = 0.5\pi$ on a uniform grid of 129×129. The convective and diffusive terms are discretized using second order centered schemes. After an initial transient phase (the fluid is initially at rest), the oscillatory flow reaches a periodic state, as Fig. 7 shows.

The temporal accuracy of the algorithm is investigated by comparing the results *during one cycle* in the periodic state. At each stage the following algorithm tolerances are imposed: $\epsilon_{solver} < 10^{-15}$, $\epsilon_{iter} < 10^{-13}$. Again 2-PISO iterations are performed in each outer iteration (j). The temporally exact solution is obtained using ESDIRK-5 scheme with $\Delta t = \mathcal{T}/1024$. The (time integration) error in the cell-center velocity and pressure are shown as a function of time step size, in respectively Fig. 8(a)

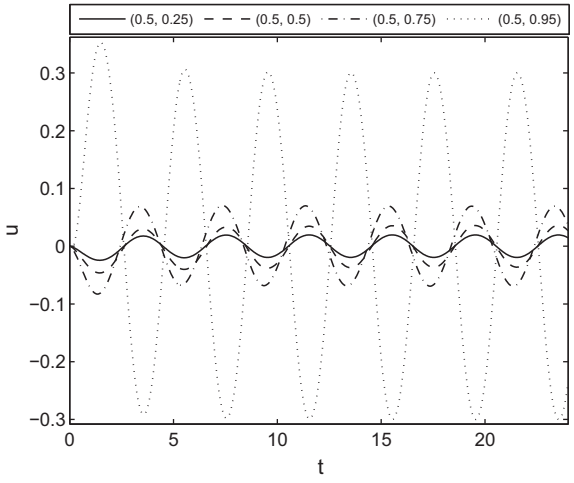

Fig. 7. Time evolution of the x-component of the velocity field during a few cycles in several locations along the midplane $x = 0.5$ for the oscillatory lid driven cavity flow. The results are obtained using ESDIRK3 with $\Delta t = \frac{\mathcal{T}}{128}$ (where $\mathcal{T} = 4.0$).

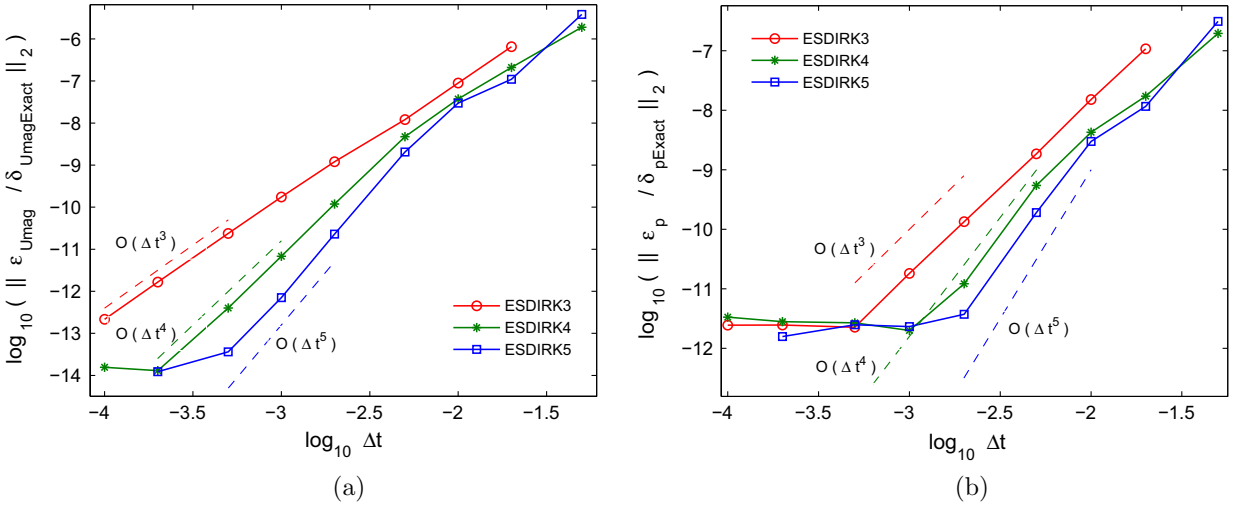

Fig. 6. Error as a function of Δt for (a) cell-center velocity field magnitude and (b) pressure field, for the lid driven cavity test case with the unstructured mesh.

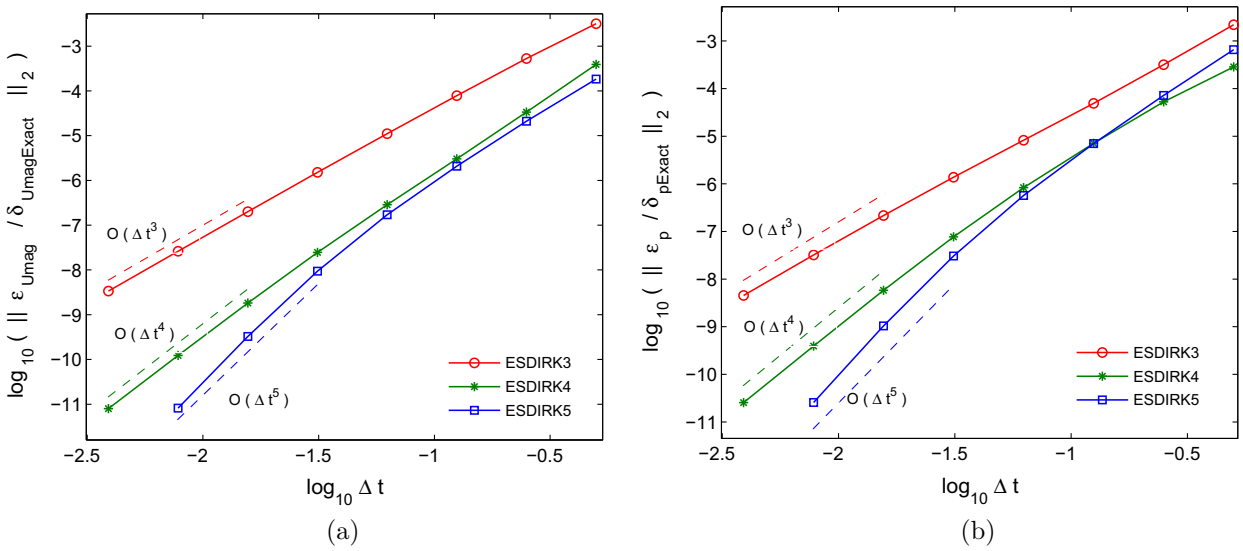

Fig. 8. Error as a function of Δt for (a) cell-center velocity field magnitude and (b) pressure field, for the oscillatory lid driven cavity test case.

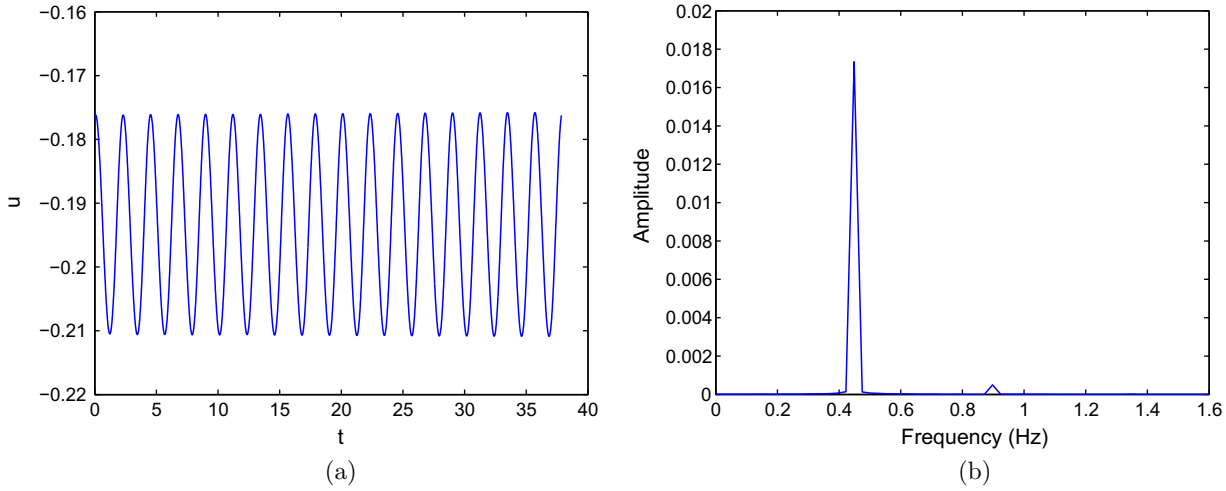

Fig. 9. (a) Time evolution of the x-component of the velocity at the monitoring point $(\frac{2}{16}, \frac{3}{16})$ and (b) the corresponding amplitude spectrum, for $Re = 8200$ lid driven cavity flow. The results are obtained using ESDIRK3 with $\Delta t = \frac{\mathcal{T}}{128}$ (where $\mathcal{T} = 2.2$).

and (b). As the results in the figures demonstrate, the temporal design order is preserved in all the flow variables in the asymptotic range. The largest Courant number during the simulations using ESDIRK-5 was $Co \approx 60$ which corresponds to 8 time-steps over one period $\Delta t = \mathcal{T}/8$, and can be considered as relatively coarse.

The number of (Picard) iterations per stage required to converge to the imposed tolerance ϵ_{iter} was found to be noticeably higher for high Courant numbers compared to low Courant numbers. One means to reduce the number of iterations is incorporating under-relaxation. In a similar manner as for to the temporal terms, care should be taken in including under-relaxation into the discretized equations in order to avoid converged solutions being dependent on the under-relaxation factor. For further details see for example [1,9,27].

In addition, in the present work, to investigate temporal order, a fixed time step has been used for the simulations. For a fixed temporal accuracy however, the time step usually varies during a typical simulation [5]. Therefore, a fixed time step may not be the optimal choice. The Runge–Kutta schemes include embedded schemes which allow the use of automatic error-based time-step controllers. Another factor which will reduce the efficiency of the algorithm is performing Picard iterations to inappropriate tolerance levels; once the iterative error (splitting error and non-linear iterative error) is adequately below the time integration error, performing additional iterations does not increase the solution accuracy, but does increase the computational cost. One the other hand, if sufficient iterations are not performed, the solution is dominated by the iterative error which will result in the deterioration of the solution accuracy and/ or instability [5,18]. An additional benefit of the error estimate is providing a suitable termination strategy for the iterations. For details and examples on the use of automatic time-step controllers the interested reader is referred to [4,3,17] (in which the Runge–Kutta schemes of [17] considered in this paper are used).

6.4. Lid driven cavity flow at $Re = 8200$

As a final test case the lid driven cavity at a Reynolds number above the critical value at which the first Hopf bifurcation occurs is considered. Unlike the previous test case, the flow is inherently unsteady for this test case. This problem has been studied for example in [28,29]. The critical Reynolds number has been reported in [28] to be close to $Re = 8000$, though this varies to some degree in the literature, mainly due to the chosen numerical

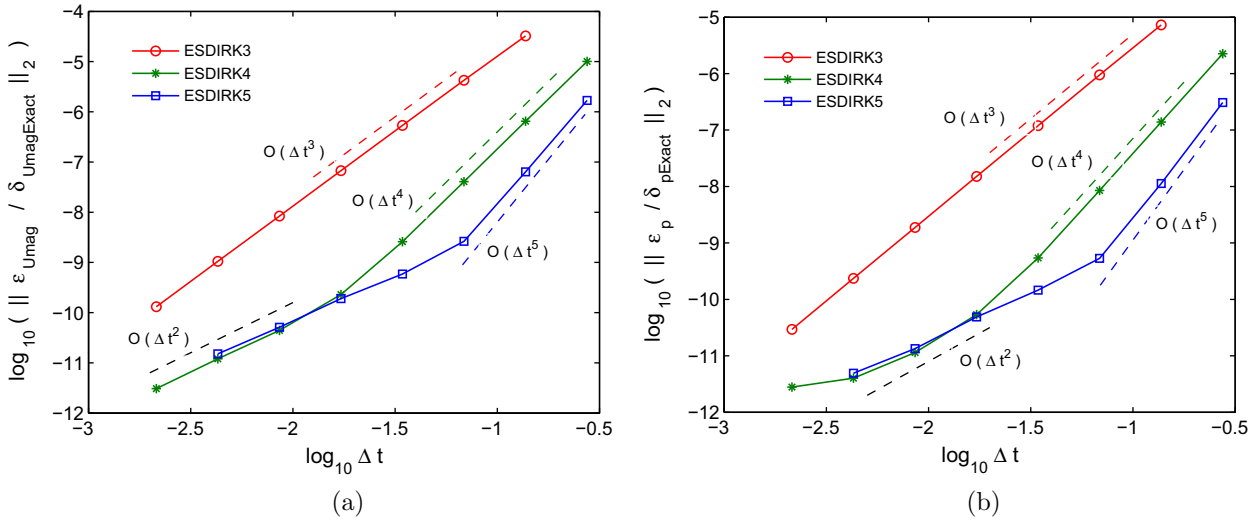

Fig. 10. Error as a function of Δt for (a) cell-center velocity field magnitude and (b) pressure field, for $Re = 8200$ lid driven cavity test case.

method [29]. Here, the lid driven cavity flow at the Reynolds number of $Re = 8200$ is considered. A non-uniform grid of 511×511 with mesh refinement near the boundaries is used in the study ($\frac{\Delta x_{max}}{\Delta x_{min}} = 2.66$ with $\Delta x_{min} = 0.001$). The convection and diffusion terms are respectively discretized using the second order linear upwind and centered schemes.

To perform the temporal order analysis, an initial solution is computed to step over the transients and arrive at the periodic state. The solution within the periodic state is then taken as the initial condition for the time step convergence analysis. The time history of the x-component of the velocity at the monitoring point $(\frac{2}{16}, \frac{3}{16})$ is shown in Fig. 9(a). The figure depicts several cycles within the periodic region. The monitoring point corresponds to the location of $(\frac{14}{16}, \frac{13}{16})$ reported in [28] in which the lid is moving to the left. Some differences with the corresponding results of [28] can be observed, such as a difference of 3% in the mean value of the oscillations and a factor of ≈1.6 larger amplitude relative to [28]. The corresponding amplitude spectrum of the monitoring point's time-history, obtained using Fourier analysis, is depicted in Fig. 9(b). The periodic solution has one dominant frequency of $f = 0.45$ (and two higher modes of much smaller amplitude, one at 0.9 and the other at 1.35) and is in agreement with the reported value in [28]. Therefore, the selected grid provides sufficient spatial resolution to capture the large-scale features of the flow. Temporal order analysis is performed based on resolving the dominant frequency of the periodic flow. The reader interested in an elaborate discussion on the flow physics of the lid driven cavity flow at such Reynolds numbers including phenomena of the transient phase is referred to [30].

Simulations are performed for one cycle ($\mathcal{T} = \frac{1}{f} = 2.2$) using ESDIRK3-ESDIRK5 with time step sizes ranging from $\Delta t = \frac{\mathcal{T}}{1024}$ to $\Delta t = \frac{\mathcal{T}}{8}$. At each stage of the ESDIRK schemes the following algorithm tolerances are imposed: $\epsilon^{\boldsymbol{U}}_{solver} < 10^{-15}$, $\epsilon^{\boldsymbol{p}}_{solver} < 10^{-12}$, and $\epsilon_{iter} < 1 \cdot 10^{-12}$. 2-PISO iterations are performed in each outer iteration. It is noted that the iterative algorithm did not yield a stable solution for $\Delta t = \frac{\mathcal{T}}{4}$ for any of the three ESDIRK schemes, as well as $\Delta t = \frac{\mathcal{T}}{8}$ for ESDIRK3. The temporally exact solution is obtained using ESDIRK5 with $\Delta t = \frac{\mathcal{T}}{1024}$. The temporal accuracy of the algorithm is investigated by comparing simulation results for the complete cycle. The (time integration) error in the cell-center velocity and pressure are shown as a function of time step size, in respectively Fig. 10(a) and (b).

As the results in the figures demonstrate, the temporal design order of the three ESDIRK schemes is observed in both the velocity and pressure for the large time step sizes. The results for the large time step sizes also correspond to large Courant numbers. For example, in the simulations of ESDIRK5 with $\Delta t = \frac{\mathcal{T}}{8}$, a maximum Courant number of approximately $Co \approx 224$ is observed. The observed convergence rate of ESDIRK5 for the smaller time step sizes, corresponding to errors less than 10^{-9}, falls below the designed order of accuracy which is possibly caused by severe stiffness in the system. A similar observation can be made for convergence rate of ESDIRK4.

7. Conclusions

Using potential gain in computational efficiency relative to commonly used second order time integration schemes as motivation, the high order ESDIRK schemes have been used for advancing the solution to the unsteady incompressible Navier–Stokes equations in time. The cell-centered collocated finite volume method has been used for spatial discretization. A face velocity interpolation procedure which preserves temporal design order of the multi-stage ESDIRK schemes has been introduced. The resultant fully discredited equation is solved using a segregated iterative time advancing algorithm (iterated-PISO) which comprises the introduced face velocity equation. The results from solving numerical examples demonstrated the temporal order preservation of the algorithm (in the absence of severe stiffness in the problem).

As an extension of the current work, application of automatic error-based time-step controllers may be considered. Efficiency relative to use of a fixed time-step size and gain from imposing a termination strategy for the stage iterations should be investigated.

Acknowledgement

The financial support provided by Netherlands Technology Foundation STW Grant Number 10113 is greatly appreciated. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. The first author is also grateful to Dr. Željko Tuković, University of Zagreb, for the fruitful discussions and valuable comments in code implementation.

References

[1] Miller T, Schmidt F. Use of a pressure-weighted interpolation method for the solution of the incompressible Navier–Stokes equations on a nonstaggered grid system. Numer Heat Transfer 1988;14:213–33.

[2] Rhie C, Chow W. Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA J 1983;21:1525–32.
[3] Kanevsky A, Carpenter MH, Gottlieb D, Hesthaven JS. Application of implicit–explicit high order Runge–Kutta methods to discontinuous-Galerkin schemes. J Comput Phys 2007;225:1753–81.
[4] Volker J, Joachim R. Adaptive time step control for the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 2010;199:514–24.
[5] Bijl H, Carpenter M, Vatsa V, Kennedy C. Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flow. J Comput Phys 2002;179:313–29.
[6] Johansson P, Davidson L. Modified collocated SIMPLEC algorithm applied to buoyancy-affected turbulent flow using a multigrid solution procedure. Numer Heat Transfer Part B 1995;28:39–57.
[7] Wesseling P. Principles of computational fluid dynamics. Springer; 2000.
[8] Cubero A, Fueyo N. A compact momentum interpolation procedure for unsteady flows and relaxation. Numer Heat Transfer, Part B 2007;52:471–93.
[9] Lien FS, Leschziner MA. A general non-orthogonal collocated finite volume algorithm for turbulent flow at all speeds incorporating second-moment turbulence–transport closure, Part 2: Application. Comput Methods Appl Mech Eng 1994;114:149–67.
[10] Pascau A. Cell face velocity alternatives in a structured colocated grid for the unsteady Navier–Stokes equations. Int J Numer Methods Fluids 2011;65:812–33.
[11] Yu B, Tao WQ, Wei JJ. Discussion on momentum interpolation method for collocated grids of incompressible flow. Numer Heat Transfer Part B 2002;42:141–66.
[12] Tuković Ž, Jasak H. A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow. Comput Fluids 2012;55:70–84.
[13] LeVeque R. Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. SIAM; 2007.
[14] Jasak H. Error analysis and estimation for finite volume method with applications to fluid flows. PhD thesis, Imperial College, University of London; 1996.
[15] Hairer E, Wanner G. Solving ordinary differential equations II: Stiff and differential–algebraic problems. 3rd ed. Springer; 2004.
[16] Sengupta TK. High accuracy computing methods: fluid flows and wave phenomena. Cambridge University Press; 2013.
[17] Kennedy CA, Carpenter MH. Additive Runge–Kutta schemes for convection–diffusion–reaction equations. Appl Numer Math 2003;44:139–81.
[18] Carpenter MH, Kennedy CA, Bijl H, Viken SA, Vatsa VN. Fourth-order Runge–Kutta schemes for fluid mechanics applications. J Sci Comput 2005;25:157–94.
[19] Najafi-Yazdi A, Mongeau L. A low-dispersion and low-dissipation implicit Runge–Kutta scheme. J Comput Phys 2013;233:315–23.
[20] Bhaumik S, Sengupta S, Sengupta A. Wave properties of fourth-order fully implicit Runge–Kutta time integration schemes. Comput Fluids 2013;81:110–21.
[21] Chang W, Giraldo F, Perot B. Analysis of an exact fractional step method. J Comput Phys 2002;180:183–99.
[22] Perot B. An analysis of the fractional step method. J Comput Phys 1993;108:51–8.
[23] Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. J Comput Phys 1986;62:40–65.
[24] Mohamad AA. Benchmark solution for unsteady state CFD problems. Numer Heat Transfer, Part A: Appl 1998;34:653–72.
[25] Iwatsu R, Hyun JM, Kuwahara K. Numerical simulation of flows driven by a torsionally oscillating lid in a square cavity. J Fluids Eng 1992;114:143–51.
[26] Soh WY, Goodrich JW. Unsteady solution of incompressible Navier–Stokes equations. J Comput Phys 1988;79:113–34.
[27] Majumdar S. Role of underrelaxation in momentum interpolation for calculation of flow with nonstaggered grids. Numer Heat Transfer 1988;13:125–32.
[28] Bruneau CH, Saad M. The 2D lid-driven cavity problem revisited. Comput Fluids 2006;35:326–48.
[29] Sengupta TK, Lakshmanan V, Vijay VVSN. A new combined stable and dispersion relation preserving compact scheme for non-periodic problems. J Comput Phys 2009;228:3048–71.
[30] Sengupta TK, Vijay VVSN, Singh N. Universal instability modes in internal and external flows. Comput Fluids 2011;40:221–35.